

Unit Testing Plan for Digital Watch System

- Test Plan
- Test Design Specification
- Test Cases Specification

Project Team

이동아

Date

2012-11-08

Team Information

이동아: dalee.dslab@gmail.com

Table of Contents

1	Introduction	4
1.1	Objectives.....	4
1.2	Background	4
1.3	Scope.....	4
1.4	Project plan	4
1.5	Configuration management plan.....	4
1.6	References.....	4
2	Test items	4
3	Approach.....	6
4	Item pass/fail criteria	6
5	Unit test design specification.....	6
5.1	Test design specification identifier	6
5.2	Features to be tested	6
5.3	Approach refinements.....	6
5.4	Test identification	6
5.5	Feature pass/fail criteria	6
6	Unit test case specification.....	6
6.1	Test case specification identifier	6
6.2	Test items	6
6.3	Input specifications.....	7
6.4	Output specifications.....	7

7	Testing tasks	7
8	Environmental needs	7
	Eclipse IDE for C/C++ developers Visual SVN server Hudson CI	7
9	Unit Test deliverables	7
10	Schedules	7

1 Introduction

1.1 Objectives

본 문서는 2012년 2학기 SE 수업의 T2가 개발한 두 번째 버전의 Digital Watch System (DWS)을 Unit Testing하기 위한 계획문서이다. T2가 정의한 Unit을 Testing을 수행하기 위하여 Testing Pass/Fail Criteria를 정의하고 이를 수행하기 위한 Test design & test cases를 제작한다.

1.2 Background

2012년 2학기 SE수업에서 개발하는 모든 DWS는 SASD기법을 이용하여 개발된다. 기능 별로 나뉜 Unit은 SRA, SDS 문서에 모두 정의되어 있다.

1.3 Scope

SRA 및 SDS 문서에 정의된 Unit을 Testing한다.

1.4 Project plan

1.5 Configuration management plan

1.6 References

T2-2012.DWS.SRA-3.1

T2-2012.DWS.SDS-2.1

2 Test items

T2가 SASD기법을 이용하여 개발한 DWS을 Testing한다. SA와 SD에서 분류한 각 process/module 별로 Testing을 수행한다. <오류! 참조 원본을 찾을 수 없습니다.>은 SA를 이용하여 요구사항을 분석한 결과 중 전체 DFD를 나타낸 그림이며, <오류! 참조 원본을 찾을 수 없습니다.>은 SD의 Basic Structural Chart를 나타낸 그림이다. 각 그림을 참조하여 Unit을 지정하고, 지정한 Unit을 SRA에 명세 된 내용과 같은 동작을 하는지 확인한다.

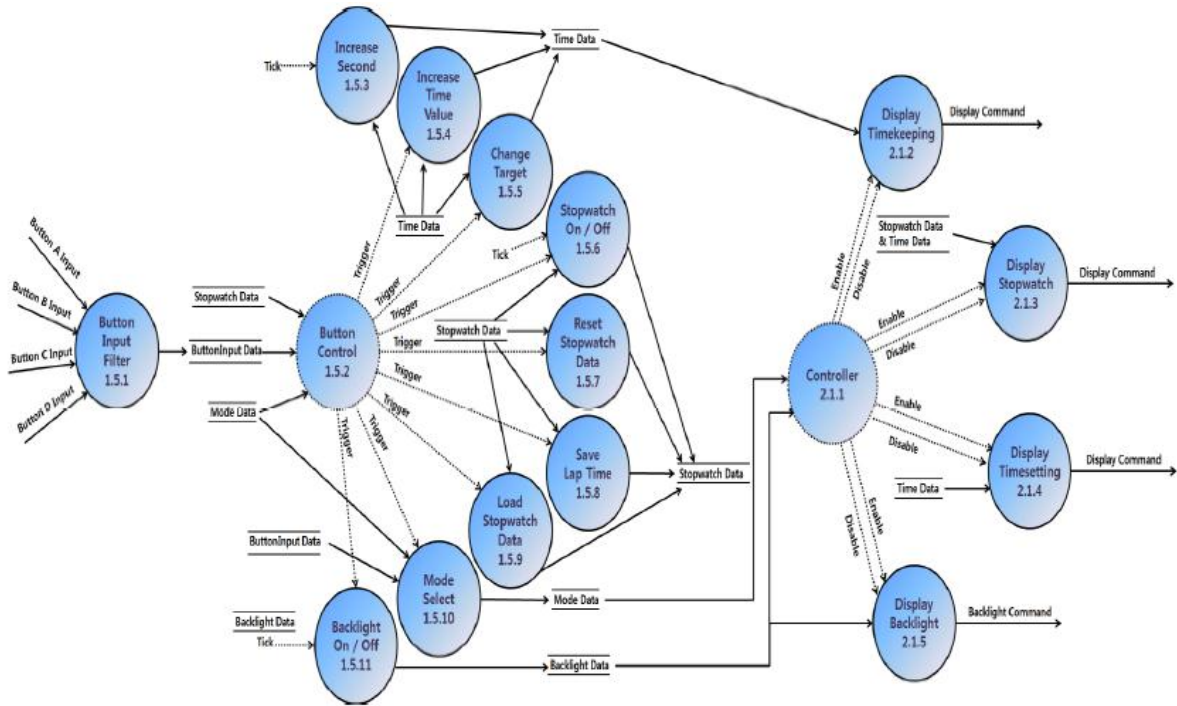


Figure 1 Overall DFD

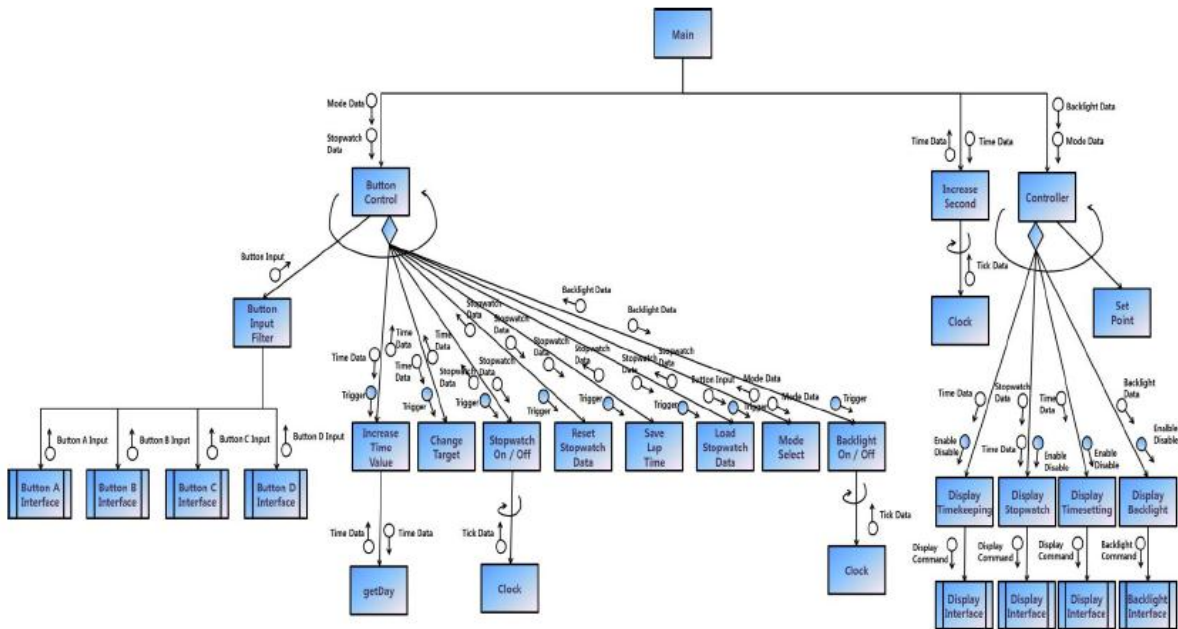


Figure 2 Structural Chart

3 Approach

Brute Force, SRA의 가장 하위 Level의 Process를 대상으로 Test를 수행한다.

4 Item pass/fail criteria

모든 Units은 SRA에 명시된 Process Specification과 동일한 결과를 출력해야 한다. 각 Unit별 Pass/Fail Criteria는 를 참조한다.

5 Unit test design specification

5.1 Test design specification identifier

DWS.UTC.Number

5.2 Features to be tested

Process 1.1-1.4

Process 2.2-2.3

Process 1.5.1-1.5.B

Process 2.1.1-2.1.5

5.3 Approach refinements

각 Process Specification에 명시된 내용을 기반으로 Test Design 및 Test Cases를 생성해 낸다.

5.4 Test identification

DWS.UTC.Number

5.5 Feature pass/fail criteria

6 Unit test case specification

6.1 Test case specification identifier

DWS.UTC.Number.Number

6.2 Test items

<Table 2 Test Design Specification> 참조

6.3 Input specifications

<Table 3 Test Case Identification> 참조

6.4 Output specifications

<Table 3 Test Case Identification> 참조

7 Testing tasks

Table 1 Test tasks & schedule

Task	Predecessor tasks	Special skills	Effort	Finish date
(1) Unit Test Plan 작성	SRA 작성 SDS 작성 DWS 구현		3	
(2) Test design specification	Task 1	DWS 에 대한 이해	5	
(3) Test case specification	Task 2	DWS 에 대한 이해	5	
(4) Test execution	Task 3	Test code 작성 Test tools 에 대한 이해	4	
(5) Test result report	Task 4		1	
(6) 개발팀에 report 전달	Task 5		1	

8 Environmental needs

Eclipse IDE for C/C++ developers

Visual SVN server

Hudson CI

9 Unit Test deliverables

Unit Test Report

10 Schedules

<Table 1 Test tasks & schedule> 참조

Table 2 Test Design Specification

Identifier	Feature	Pass/Fail
DWS.UTC.110	Button A Input을 받아 Button A Input에 출력	출력 확인
DWS.UTC.120	Button B Input을 받아 Button B Input에 출력	출력 확인
DWS.UTC.130	Button C Input을 받아 Button C Input에 출력	출력 확인
DWS.UTC.140	Button D Input을 받아 Button D Input에 출력	출력 확인
DWS.UTC.220	Display Command를 받아서 Display에 출력	출력 확인
DWS.UTC.230	Backlight Command를 받아서 Backlight 출력	출력 확인
DWS.UTC.151	Button A-D Input을 받아서 ButtonInput Data에 저장 D(4)>C(3)>B(2)>A(1) 우선순위로 하나만 저장	입력 값 저장 확인 우선순위 확인
DWS.UTC.152	STD 동작확인	
DWS.UTC.153	100 Tick 일 때 초를 증가시킴 60초가 되면 0으로 초기화하며 분을 증가	초 증가 확인 초 증가만으로 시간이 잘 흐르는지 확인
DWS.UTC.154	Target이 가리키는 Time Data를 1 증가 Target이 가질 수 있는 값의 범위에 따른 변화	Target이 가리키는 값이 정상적으로 변하는지 확인 각 값의 범위에 따른 변화 확인
DWS.UTC.155	Target 값 변화	0-5까지 변화
DWS.UTC.156	Stopwatch 정지 중; Stopwatch 동작 Stopwatch 동작 중; Stopwatch 정지	동작데이터 변화 확인 0 → 1 1 → 0
DWS.UTC.157	Stopwatch의 측정시간 초기화	초기화 여부 확인
DWS.UTC.158	Stopwatch의 측정시간을 Lapttime에 저장 및 Lapttime 출력여부 값 변경	Lapttime 저장여부 및 출력여부 값 변화 확인 0→1

DWS.UTC.159	Laptime 출력여부 데이터 변경	1→0
DWS.UTC.15A	Modedata == 1 && Button A Modedata == 1 && Button C Modedata == 2 && Button C Modedata == 3 && Button A	1→3 1→2 2→1 3→1
DWS.UTC.15B	Backlight == 0 Backlight == 1	Backlight == 1 200Tick 후에 Backlight == 0
DWS.UTC.211	STD 동작 확인	
DWS.UTC.212	Time Data를 Display Command로 출력	출력 확인
DWS.UTC.213	Time Data와 Stopwatch data를 Display Command로 출력	출력 확인
DWS.UTC.214	Time Data를 Display Command로 출력	출력 확인
DWS.UTC.215	Backlight Data == 0 Backlight Data == 1	Backlight Command == off Backlight Command == on

Table 3 Test Case Identification

Identifier	Input	Output	비고
DWS.UTC.110.00	System 함수 호출	PASS	<code>getch()</code>
DWS.UTC.120.00	System 함수 호출	PASS	<code>getch()</code>
DWS.UTC.130.00	System 함수 호출	PASS	<code>getch()</code>
DWS.UTC.140.00	System 함수 호출	PASS	<code>getch()</code>
DWS.UTC.220.00	System 함수 호출	PASS	<code>printf()</code>
DWS.UTC.230.00	System 함수 호출	PASS	<code>system("cmd.exe")</code>
DWS.UTC.151.00	여러 개의 입력을 받을 수 없는 구조		<code>buttonInputFilter()</code>
DWS.UTC.152.00	State 정보를 찾을 수 없음	FAIL	<code>void buttonControl()</code>
DWS.UTC.153.00	100 Tick && Time Data = 12:34:56	Time Data = 12:34:57	<code>void increaseSecond()</code>
DWS.UTC.153.01	100 Tick && Time Data = 12:34:59	Time Data = 12:35:00	
DWS.UTC.153.02	100 Tick && Time Data = 12:59:59	Time Data = 13:00:00	
DWS.UTC.153.03	100 Tick && Time Data = 20-11-22 23:59:59	Time Data = 20-11-23 00:00:00	
DWS.UTC.153.04	100 Tick && Time Data = 20-11-30 23:59:59	Time Data = 20-12-01 00:00:00	
DWS.UTC.153.05	100 Tick && Time Data = 20-12-31 23:59:59	Time Data = 21-01-01 00:00:00	
DWS.UTC.153.06	100 Tick && Time Data = 20-02-28 23:59:59	Time Data = 20-02-29 00:00:00	
DWS.UTC.153.07	100 Tick && Time Data = 21-02-28 23:59:59	Time Data = 21-03-01 00:00:00	
DWS.UTC.154.00	Target == 0 && Time Data = 20-11-22 23:59:00	Time Data = 20-11-22 23:59:01	<code>void increaseTimeValue()</code>
DWS.UTC.154.01	Target == 0 && Time Data = 20-11-22 23:59:59	Time Data = 20-11-22 23:59:00	
DWS.UTC.154.02	Target == 1 && Time Data = 20-11-22 20:00:00	Time Data = 20-11-22 21:00:00	
DWS.UTC.154.03	Target == 1 && Time Data = 20-11-22 23:00:00	Time Data = 20-11-22 00:00:00	
DWS.UTC.154.04	Target == 2 && Time Data = 20-11-22 23:00:00	Time Data = 20-11-22 23:01:00	
DWS.UTC.154.05	Target == 2 && Time Data = 20-11-22 23:59:00	Time Data = 20-11-22 23:00:00	

DWS.UTC.154.06	Target == 3 && Time Data = 20-11-22 23:00:00	Time Data = 21-11-22 23:00:00	
DWS.UTC.154.07	Target == 3 && Time Data = 99-11-22 23:00:00	Time Data = 12-11-22 23:00:00	
DWS.UTC.154.08	Target == 4 && Time Data = 20-11-22 23:00:00	Time Data = 20-12-22 23:00:00	
DWS.UTC.154.09	Target == 4 && Time Data = 20-12-22 23:00:00	Time Data = 20-01-22 23:00:00	
DWS.UTC.154.11	Target == 5 && Time Data = 20-12-22 23:00:00	Time Data = 20-12-23 23:00:00	
DWS.UTC.154.12	Target == 5 && Time Data = 20-12-31 23:00:00	Time Data = 20-12-01 23:00:00	
DWS.UTC.154.13	Target == 5 && Time Data = 20-02-28 23:00:00	Time Data = 20-02-01 23:00:00	
DWS.UTC.154.14	Target == 5 && Time Data = 21-02-28 23:00:00	Time Data = 21-02-29 23:00:00	
DWS.UTC.154.15	Target == 5 && Time Data = 21-02-29 23:00:00	Time Data = 21-02-01 23:00:00	
DWS.UTC.155.00	Target == 0	Target == 1	void changeTarget()
DWS.UTC.155.01	Target == 5	Target == 0	
DWS.UTC.156.00	Stopwatch.runningcondition == 0	Stopwatch.runningcondition == 1	void stopwatchOnOff()
DWS.UTC.156.01	Stopwatch.runningcondition == 1	Stopwatch.runningcondition == 0	
DWS.UTC.157.00		Stopwatch.time == 00:00:00	void resetStopwatchData()
DWS.UTC.158.00	Stopwatch.Time == 12:34:56 && Stopwatch.Lapdisplay == 0	Stopwatch.Laptime == 12:34:56 && Stopwatch.Lapdisplay == 1	void saveLaptime()
DWS.UTC.159.00	Stopwatch.Lapdisplay == 1	Stopwatch.Lapdisplay == 0	void loadStopwatchData()
DWS.UTC.15A.00	Modedata == 1 && Button A	Modedata == 3	void modeSelect(int button)
DWS.UTC.15A.01	Modedata == 1 && Button C	Modedata == 2	
DWS.UTC.15A.02	Modedata == 2 && Button C	Modedata == 1	
DWS.UTC.15A.03	Modedata == 3 && Button A	Modedata == 1	
DWS.UTC.15B.00	BacklightData == 0	BacklightData == 1	void backlightOnOff()
DWS.UTC.15B.01	BacklightData == 1	BacklightData == 0	
DWS.UTC.211.00	State정보를 찾을 수 없음	FAIL	void controller()

DWS.UTC.212.00	System 함수 호출	PASS	void displayTimekeeping()
DWS.UTC.213.00	System 함수 호출	PASS	void displayStopwatch()
DWS.UTC.214.00	System 함수 호출	PASS	void displayTimesetting()
DWS.UTC.215.00	System 함수 호출	PASS	void displayBacklight()